

ESO universal Horn and LFP logics: separation between machine level and structure level

P. Manyem
 Department of Mathematics
 Shanghai University
 Shanghai 200444 China.
 Email: prabhu.manyem@gmail.com

January 15, 2012

Abstract

Consider decision problems derived from optimization problems. For problems in this category, we show that they cannot be expressed in existential second order (ESO) logic with a first order part that is universal Horn (or simply, ESO Π_1 Horn) at the structure level, as opposed to machine level. This is true even when we consider ordered structures. We show that an **NP**-complete problem (Maximum Independent Set) and a polynomially solvable problem (Maximum Matching) have identical ESO Π_1 (non-Horn) expressions when the only unknowns are the second order predicates, hence this logic is too weak to distinguish between the two classes. At the structure level, Fagin's theorem is false in one sense, in that ESO universal sentences under ordered structures cannot capture the class **NP**. Also at the structure level, under ordered structures, least fixed point logic with counting (LFP+C) is insufficient to capture **P**, the class of problems decidable in polynomial time.

Keywords. Descriptive complexity, Optimization, Existential second order logic, Least fixed point logic, Machine level and structure level.

AMS classification. 90C99, 68Q19, 68Q15, 68Q17, 03C13.

1 Introduction

We first introduce some notation and definitions. For the contributions of this article, see Sec. 1.2.

We work with decision versions of optimization problems. The reader is assumed to have some background in Finite Model Theory. If not, the book by Ebbinghaus and Flum [4] serves as a good introduction. Publications [8] and [13] are also relevant material for this line of research.

Definition 1. Let P (NP) be the class of problems decidable in time polynomial in the size of the input deterministically (non-deterministically), respectively.

Definition 2. [16] A **P-optimization** problem Q is a tuple $Q = \{I_Q, F_Q, f_Q, \text{opt}_Q\}$, where

- (i) I_Q is a set of instances to Q ,
- (ii) $F_Q(I)$ is the set of feasible solutions to instance I ,
- (iii) $f_Q(I, S)$ is the objective function value to a solution $S \in F_Q(I)$ of an instance $I \in I_Q$. It is a function $f : \bigcup_{I \in I_Q} [\{I\} \times F_Q(I)] \rightarrow \mathbb{R}_0^+$ (non-negative reals)¹, computable in time polynomial in the size $|A|$ of the domain A of I^2 ,
- (iv) For an instance $I \in I_Q$, $\text{opt}_Q(I)$ is either the minimum or maximum possible value that can be obtained for the objective function, taken over all feasible solutions in $F_Q(I)$.

$$\text{opt}_Q(I) = \max_{S \in F_Q(I)} f_Q(I, S) \text{ (for P-maximization problems),}$$

$$\text{opt}_Q(I) = \min_{S \in F_Q(I)} f_Q(I, S) \text{ (for P-minimization problems),}$$
- (v) The following decision problem is in the class **P**: Given an instance I and a non-negative constant K , is there a feasible solution $S \in F_Q(I)$, such that $f_Q(I, S) \geq K$ (for a P-maximization problem), or $f_Q(I, S) \leq K$ (in the case of a P-minimization problem)?
And finally,
- (vi) An optimal solution $S_{\text{opt}}(I)$ for a given instance I can be computed in time polynomial in $|I|$, where $\text{opt}_Q(I) = f_Q(I, S_{\text{opt}}(I))$. ■

The set of all such P-optimization problems is the **P_{opt}** class.

A similar definition, for NP-optimization problems, appeared in Panconesi and Ranjan (1993) [17]:

Definition 3. An **NP-optimization** problem is defined as follows. Points (i)-(iv) in Def. 2 above apply to NP-optimization problems, whereas (vi) does not. Point (v) is modified as follows:

- (v) The following decision problem is in **NP**: Given an instance I and a non-negative constant K , is there a feasible solution $S \in F_Q(I)$, such that $f_Q(I, S) \geq K$ (for an NP-maximization problem), or $f_Q(I, S) \leq K$ (in the case of an NP-minimization problem) ?

The set of all such NP-optimization problems is the **NP_{opt}** class, and **P_{opt}** \subseteq **NP_{opt}**.

Definition 4. Decision versions.

See Def. 2. Given a non-negative constant K and an instance $I \in I_Q$, the decision version of a P-optimization problem Q asks whether there is a feasible solution $S \in F_Q(I)$, such that $f_Q(I, S) \geq K$ (if Q is a maximization problem), or $f_Q(I, S) \leq K$ (if Q is a minimization problem).

Similarly, decision versions of NP-optimization problems are derived from Def. 3. ■

¹Of course, when it comes to computer representation, rational numbers will be used.

²Strictly speaking, we should use $|I|$ here, where $|I|$ is the length of the representation of I . However, $|I|$ is polynomial in $|A|$, hence we can use $|A|$.

As we will see in Sections 2.3–2.5, the decision versions can be expressed as a conjunction of an objective function constraint (OFC) and basic feasibility constraints (BFC).

Assumption 5. *We consider ordered structures only.*

Definition 6. *ESO Π_1 Horn* is defined as existential second order (ESO) logic where the first order part is a universal Horn formula,

We need the following result. See [7], or Theorem 3.2.17 (Page 147) in [8], or Theorem 9.32 (Page 154) in [13].

Theorem 7. *If a decision problem P_1 can be represented by an ESO Π_1 Horn sentence, then the second order quantifiers can be determined in polynomial time (PTIME). Hence P_1 can be solved in PTIME.*

This follows from the fact that HornSAT (satisfiability of a propositional Horn formula) can be solved in PTIME³ [15].

The goal is to determine values for the second order quantifiers (the unknowns in the optimization problem) in PTIME; this would imply that the problem represented by the existential second order (ESO) sentence is solvable in PTIME.

Assumption 8. *We work only with undirected graphs. If there is an edge between vertices u and v where $u < v$, we set $E(u, v) \equiv \text{true}$ and $E(v, u) \equiv \text{false}$.*

1.1 More about BFC and OFC

We represent decision versions of optimization problems as a conjunction of a single objective function constraint (OFC) and a set of basic feasibility constraints (BFC). The motivation for this comes from a general mathematical programming framework where optimization problems are expressed in the form:

$$\begin{aligned} &\text{Maximize (or Minimize) } f(\mathbf{x}), \\ &\text{subject to the following constraints:} \\ &g_i(\mathbf{x}) \leq 0, \quad 1 \leq i \leq m, \\ &h_j(\mathbf{x}) = 0, \quad 1 \leq j \leq p, \\ &\text{and } \mathbf{x} \in X \text{ (for example, } X = \mathbb{R}^n\text{).} \end{aligned} \tag{1}$$

In (1), $f(\mathbf{x})$ is the *objective function*. Additionally, for the decision problem, we are given a constant K , a non-negative integer.

Above, the BFC comprises the $m + p$ constraints $g_i(\mathbf{x}) \leq 0$ and $h_j(\mathbf{x}) = 0$, and the OFC comprises the single constraint $f(\mathbf{x}) \geq K$ for maximization problems ($f(\mathbf{x}) \leq K$ for minimization problems).

From Theorem 7 and the fact that decision versions of optimization problems can be expressed as conjunctions of OFC and BFC, one would expect that for problems in the class \mathbf{P} , the

³In this paper, we only use PTIME as a shorthand for polynomial time, not to represent the problems in the class \mathbf{P} .

OFC and the BFC can be expressed as ESO Π_1 Horn sentences. However, in Sec. 4, we will show that this is not true. In addition, we observed the following:

BFC. Interestingly, we found that the BFC for some **NP**-complete problems (Maximum Clique, Maximum Independent Set and MaxSAT) can also be expressed as ESO Π_1 Horn sentences⁴. Gate and Stewart [6] also observed this phenomenon when they derived a BFC in ESO Π_1 Horn form for another **NP**-complete problem, MaxHorn2SAT.

OFC. Decision versions of optimization problems, no matter what their complexity is, can be grouped according to the arity of the second order predicate (say F) which measures the objective function value. The objective function value is equal to $|F|$. For maximization (minimization) problems, the OFC is $|F| \geq K$ ($|F| \leq K$) respectively, where K is part of the input.

For instance, we will see in Section 2 that a problem in **P** (Maximum Matching) and two **NP**-complete problems (Maximum Clique and Maximum Independent Set) have identical OFC's, since the arity of the objective function predicate F is equal for all three problems (equal to one).

1.2 Our contribution

A few observations are in order:

- Descriptive Complexity is a very useful tool at the structure level, *not* at the machine level.
- A large subset of problems in the class **NP** is derived from optimization problems, hence this is an important class.

To elaborate on the first item above: The usefulness of Descriptive Complexity lies in the fact that if one could write a structure level expression in a certain logic for a decision problem \mathcal{D} , then it can be deduced that \mathcal{D} requires a certain amount of resource (time and/or space) for its computation; that is, we can recognize the complexity class that \mathcal{D} belongs to.

However, our main result shows that even though a large class of problems (those derived from optimization) is PTIME solvable, they cannot be expressed in ESO universal Horn logic at the structure level. Hence for such problems, the usefulness of Descriptive Complexity is lost⁵, using this particular logic.

Section 2: We explain the distinction between expressions in ESO Π_1 Horn logic at the *machine level* and the *structure level*. We provide structure level expressions for the BFC's of two problems in the class **NP**, and one in the class **P** (the corresponding optimization versions are in classes **NP_{opt}** and **P_{opt}** respectively). We observe that at the structure level, a problem in **P**, Maximum Matching, and an **NP**-complete problem, Maximum Independent Set, are identical (same signature and same expressions for the properties that need to be satisfied).

⁴In general, it is the *combination* of the BFC and OFC that makes a problem easy or difficult to solve, not just one of them.

⁵Of course, we fully acknowledge contributions in the literature that provided machine level expressions — but that is only one side of the story.

This means that ESO Π_1 sentences are unable (too weak) to distinguish between \mathbf{P} and \mathbf{NP} -complete problems at the structure level.

Section 3: We provide an expression for the OFC when the arity of the predicate measuring the objective function value is one. This sentence is in ESO Π_1 form; however, it is not a Horn sentence. (The OFC cannot be expressed as an ESO Π_1 Horn sentence; see Remark 15.)

Section 4: Theorem 20 is the main contribution of this paper.

For the class \mathbf{P} , under ordered structures, we will demonstrate that machine level expressions are more powerful. Every property in \mathbf{P} can be expressed at the machine level in ESO Π_1 Horn logic (this was proved in [7]), but there is at least one property in \mathbf{P} , Maximum Matching, for which this is not true at the structure level (proved here).

Assuming that $\mathbf{P} \neq \mathbf{NP}$, we will show that another problem decidable in PTIME, finding a solution for Vertex Cover within an approximation bound of two, cannot be expressed in ESO Π_1 Horn logic.

We also provide a proof using direct products that the upper bound form of the OFC ($|F| \leq K$) cannot be expressed in ESO Π_1 Horn logic (Theorem 27).

Section 5 (Implications for the class \mathbf{NP}): Under ordered structures, at the structure level, we observe that an important subclass of \mathbf{NP} cannot be expressed in ESO logic where the first order part is a universal formula. Also at the structure level, it is still an open problem as to whether ESO logic characterizes the class \mathbf{NP} .

Section 6 (LFP logic): The results in this section show that least fixed point logic with counting (LFP+C) can express problems in \mathbf{P} at the machine level, but insufficient to capture this class at the (more important) structure level.

We only work with structure level expressions in this paper.

2 Expressions for decision problems

2.1 Expressions at the machine level and the structure level

In this subsection⁶, we distinguish between logically expressing a problem at the (lower) *machine level* and the (higher) *structure level*. In Section 4, we will show that there is a gap (hierarchy) between these two levels for certain problems in class \mathbf{P} , if the problem is a decision version of an optimization problem.

Machine level expressions (MLE). These express the *computation* of a property (in a Turing machine, for instance). Here, *computation of a property* is defined as *checking whether a property is true using a computation device*, for a given input.

Such an expression encodes the computation steps of a Turing machine (TM), for example, as in the proof by Fagin [5] that ESO logic captures the class \mathbf{NP} and the proof by Immerman [12] that LFP (under ordered structures) captures the class \mathbf{P} . These expressions state that (i) if we go through the steps of a computation device such as a TM, and

⁶We thank Anuj Dawar for his input here.

(ii) if the TM finally reaches an accepting state,
then the property must be true for the given input structure.

For example, to write a machine level expression for *Hamiltonian Cycle* (HC), we should first design a TM that solves HC, then encode the steps of the TM in an expression.

Structure level expressions (SLE). On the other hand, structure level logical expressions do not express the *computation* of a property; rather, they directly express the truth of a property of a mathematical structure. *No computation is involved in such a description.*

(From a mathematician's viewpoint, SLE's are the natural choice to express a problem/property, *not* the MLE's. To logically express a property of a given mathematical structure, what is the need to put it through computation first?)

Properties. What is a *property*? Examples of properties would include:

- (i) existence of a Hamiltonian cycle (the input is a graph), or
- (ii) satisfiability (the input is a formula in CNF), or
- (iii) the existence of a matching of size at least K (the input being a graph and the number K).

For the purposes of this article, we formally define a property as:

Definition 9. A *property* is an isomorphism-closed class of finite structures.

In terms of a formula τ , it can be defined as:

Definition 10. A *property* \mathcal{P} is defined by a formula τ as long as the following is true: given an (input) structure A , $A \models \tau$ iff A has property \mathcal{P} .

The above definition of a *property* fits in very well with *structure level* expressions, because a formula is exactly the structure level description of a property.

Definition 11. A *decision problem* $P(\mathcal{P}, A)$ checks whether property \mathcal{P} is satisfied by a structure A . If \mathcal{P} is defined by a formula τ , then $P(\mathcal{P}, A)$ checks whether $A \models \tau$.

Henceforth, *problem* will refer to a *decision problem* unless we state that it is an optimization problem.

Machine level expressions for the class **NP** (class **P**) were provided by Fagin [5] (Grädel [7]) respectively. (The expressions for **P** assume the presence of a successor relation as a part of the first order vocabulary.)

In this article, we show that for *structure level* expressions, ESO Π_1 Horn is insufficient to capture the class **P** even when the structures are assumed to have a linear order, if the property represents the decision version of an optimization problem.

The difference between the expressions for machine level and the structure level has been pointed out earlier. See, for instance, Abiteboul, Vardi and Vianu [1]. Henceforth, we work with structure level expressions in this article.

2.2 Signature for the input structure

The standard signature for graphs is expanded to accommodate the expressions for the single OFC in each problem. The BFC for each problem will be illustrated in Secs. 2.3-2.5.

A three-sorted input structure (V, K, N) will be used. V will represent the set of vertices, K the lower bound on the objective function value, and N represents the set of natural numbers $0 \leq i \leq K$. We need two switch relations $num(x)$ and $vertex(x)$; then $num(x)$ is true iff $x \in N$, and $vertex(x)$ is true iff $x \in V$. Then a variable x has the interpretation $x^{(A)} = K$ by a structure A iff $A \models \neg num(x) \wedge \neg vertex(x)$.

Since we work with ordered structures, we assume that $V < K < N$ and hence we only consider structures A such that

- $A \models \forall u \forall v [vertex(u) \wedge \neg vertex(v) \wedge \neg num(v)] \rightarrow (u < v)$
(This means, if u is a vertex and v is interpreted to be the lower bound K , then $u < v$);
and
- $A \models \forall u \forall v [num(v) \wedge \neg vertex(u) \wedge \neg num(u)] \rightarrow (u < v)$
(If $v \in N$ and u is interpreted as K , then $u < v$).

The sub-domains V and N are assumed to be ordered. As usual, the binary relation E represents edges of the graph.

We need a few constants: min and max refer to the first and last elements in V respectively. We introduce another constant called “base” to the input vocabulary, such that $base < min$, where min is the “first” vertex. These constants will be used in Sec. 3 (appendix).

2.3 Maximum Clique: Decision Version

Definition 12. *Maximum Clique problem (decision version).*

Given a graph $G = (V, E)$ and a constant K , is it possible to mark vertices in V in such a way that

- (i) BFC: For any $x, y \in V$ where $x < y$, if x and y are marked, then $(x, y) \in E$, and
- (ii) OFC: The number of marked vertices is at least K ?

(A note about K. Readers may ask, “How is K represented in the input? in binary or unary?”. Our answer: This may be relevant for problems with large data, such as Maximum Flow or Minimum Cost Flow, where edge weights are part of the input. However, the issue is irrelevant for problems such as Maximum Clique where the solution value lies in the $[0, n]$ range, where n is the number of vertices. If $K > n$, then the solutions are immediately disqualified — there cannot be a solution to the Maximum Clique problem with a clique size more than the number of vertices. If $K \leq n$, the input size for K is at most that of n , whether in binary or unary.)

For simplicity, assume the following:

Assumption 13. $K \leq n$, where n is the number of vertices in the input graph (or in general, the domain size).

Let F be the set of marked vertices. We can then express the problem in ESO (existential second order) logic:

$$\phi \equiv \exists F (\phi_1 \wedge \phi_2 \wedge \phi_3), \text{ where} \quad (2)$$

$$\phi_1 \equiv \forall u \forall v [F(u) \wedge F(v) \wedge (u < v)] \rightarrow E(u, v), \quad \phi_2 \equiv \forall w F(w) \rightarrow \text{vertex}(w), \quad (3)$$

and ϕ_3 is an ESO sentence which captures the OFC, which is $|F| \geq K$ (the lower bound on the objective function value). Also, ϕ_2 requires that $w \in V$ (that is, $w \neq K$ and $w \notin N$) if it is to be counted towards the objective function.

ϕ_1 captures the BFC; it says that if two vertices are marked, then there is an edge between them in the input graph $G = (V, E)$.

Rewrite the above in ESO Π_1 Horn form as:

$$\phi \equiv \exists F (\phi_1 \wedge \phi_2 \wedge \phi_3), \text{ where} \quad (4)$$

$$\phi_1 \equiv \forall u \forall v \neg F(u) \vee \neg F(v) \vee \neg(u < v) \vee E(u, v), \quad \phi_2 \equiv \forall w \neg F(w) \vee \text{vertex}(w). \quad (5)$$

Of all the disjuncts in (5), except for $F(u)$, $F(v)$ and $F(w)$, the rest are first order; their truth values can be computed from the input and substituted into the formula.

Remark 14. *The Horn condition applies only to the second order predicate F and to other second order predicates created to capture the OFC ($|F| \geq K$).*

The above remark also applies to the OFC's for Maximum Matching and the Maximum Independent Set problems (see the next two subsections).

Remark 15. *The OFC, $|F| \geq K$, cannot be captured in ESO Π_1 Horn logic⁷, even in the presence of ordering. (Thanks to Anuj Dawar and Wilfrid Hodges for communicating this information.) This can be shown using the fact that universal sentences are preserved under substructures. Also note that if a property fails to hold under the ordered structures assumption, it fails to hold under the “successor relation” assumption as well, since the former assumption is stronger and subsumes the latter. ■*

As we see below, the BFC sentences for Maximum Clique (an **NP**-complete problem) and Maximum Matching (a problem in **P**) are almost identical. They become identical after we substitute the truth values for the first order predicates.

2.4 Maximum Matching: Decision Version

For this problem, we find it more convenient to let the universe for the input structure be the edge set E . We use a binary vertex relation $W(u, v)$ to indicate that there is a common vertex between edges u and v . Hence we represent a graph by $G = (E, W)$ where E is the edge set and W is the binary relation between the edges just described. We call this the *line graph* version of Maximum Matching.

Another way to think about this: $G_2 = (E, W)$ is the *line graph* of a given graph $G_1 = (V, E)$. In general, $|W|$ need **not** be equal to $|V|$; however, $|W|$ is bound by a polynomial in $|V|$.

⁷Indeed, if the OFC can be expressed in ESO Π_1 Horn, then it would mean that the decision version of Maximum Clique can also be expressed in ESO Π_1 Horn, hence this problem would be polynomially solvable.

Given a line graph G_2 , can the original graph G_1 be uniquely re-constructed? The answer is yes, as long as $|V| \geq 5$. This result is due to Whitney [18] (communicated to us by Sanming Zhou).

Due to Assumption 5, we let

$$\alpha_0 \equiv [W(u, v) \wedge \text{vertex}(u) \wedge \text{vertex}(v)] \rightarrow (u < v). \quad (6)$$

Note: Though in reality, u and v are edges, we still let $\text{vertex}(u) = \text{vertex}(v) = \text{TRUE}$ so as to use the signature defined at the beginning of this section.

Definition 16. Maximum Matching, decision problem (line graph version).

Given a graph $G = (E, W)$ and a constant K , is it possible to mark (or match) edges in E in such a way that

- (i) BFC: For any two edges $u, v \in E$, $u < v$, if u and v are adjacent, i.e. $W(u, v)$ is true, then at least one of them must **not** be marked, and
- (ii) OFC: The number of marked edges is at least K ?

Let F be the set of marked (or matched) edges. We can express the problem in ESO logic:

$$\alpha \equiv \exists F (\alpha_0 \wedge \alpha_1 \wedge \alpha_2 \wedge \alpha_3), \text{ where} \quad (7)$$

$$\alpha_1 \equiv \forall u \forall v [F(u) \wedge F(v) \wedge (u < v)] \rightarrow \neg W(u, v), \quad \alpha_2 \equiv \forall w F(w) \rightarrow \text{vertex}(w), \quad (8)$$

and α_3 is an ESO sentence which captures the OFC, which is $|F| \geq K$.

α_1 states that if edges u and v are matched, they do not share a common vertex. Note that α_0 is a first order expression, hence its truth value can be substituted into α .

We can rewrite the above in ESO Π_1 Horn form:

$$\alpha \equiv \exists F (\alpha_0 \wedge \alpha_1 \wedge \alpha_2 \wedge \alpha_3), \text{ where} \quad (9)$$

$$\alpha_1 \equiv \forall u \forall v \neg F(u) \vee \neg F(v) \vee \neg(u < v) \vee \neg W(u, v), \quad \alpha_2 \equiv \forall w \neg F(w) \vee \text{vertex}(w). \quad (10)$$

Of all the disjuncts in (10), except for $F(u)$, $F(v)$ and $F(w)$, the rest are first order; their truth values can be computed from the input and substituted into the formula. The Horn condition applies only to the second order predicate F and to other second order predicates created to capture the OFC ($|F| \geq K$).

2.5 Maximum Independent Set: Decision version

Definition 17. Maximum Independent Set or MIS (decision version).

Given a graph $G = (V, E)$ and a constant K , is it possible to mark vertices in V in such a way that

- (i) BFC: For any $x, y \in V$ where $x < y$, if x and y are marked, then $(x, y) \notin E$, and
- (ii) OFC: The number of marked vertices is at least K ?

As we will observe, at the structure level, MIS and the line graph version of Maximum Matching are one and the same problem once the truth values of the first order predicates are substituted.

Let the domain be the set V of vertices and let relation E define the set of edges. (That is, we represent the given graph, not the line graph.) We express the MIS problem in ESO logic:

$$\beta \equiv \exists F (\beta_1 \wedge \beta_2 \wedge \beta_3), \text{ where} \quad (11)$$

$$\beta_1 \equiv \forall u \forall v [F(u) \wedge F(v) \wedge (u < v)] \rightarrow \neg E(u, v), \quad \beta_2 \equiv \forall w F(w) \rightarrow \text{vertex}(w), \quad (12)$$

and β_3 is an ESO sentence which captures the OFC, which is $|F| \geq K$.

Rewrite the above in ESO Π_1 Horn form:

$$\beta \equiv \exists F (\beta_1 \wedge \beta_2 \wedge \beta_3), \text{ where} \quad (13)$$

$$\beta_1 \equiv \forall u \forall v \neg F(u) \vee \neg F(v) \vee \neg(u < v) \vee \neg E(u, v), \quad \beta_2 \equiv \forall w \neg F(w) \vee \text{vertex}(w). \quad (14)$$

We note that the BFC for the Maximum Independent Set (MIS) problem is exactly in the same format as (10), once W is replaced by E .

The second order predicate for the MIS problem is F , and E is a first order relation.

Discussion. Yet again we observe a strong resemblance between the BFC expressions for an **NP**-complete problem (Maximum Independent Set) and a problem solvable in polynomial time (Maximum Matching). Recall that Maximum Clique is also **NP**-complete.

After a transformation to the line graph in the case of Maximum Matching: We observe that the BFC expressions (5), (10) and (14) are very similar. What matters the most, the expressions involving the S.O. predicates are the same. After we substitute the truth values for the first order predicates, the three expressions become identical.

In a *model-checking* problem, we are given a structure A and a formula ϕ , and we are asked whether $A \models \phi$.

Remark 18. *At the structure level, MIS and the line graph version of Maximum Matching (Def. 16) are one and the same problem. They have the same input signature and identical ESO expressions (9–10) and (13–14). Hence the model-checking problem related to the problems in Def. 16 and 17 are the same.*

And coming to the OFC:

Remark 19. *The OFC expression will be identical for all the three problems considered so far, since the arity of F is the same (i.e. unary) in all three cases.*

All three problems can be expressed as a conjunction of their respective BFC and OFC. From this, we can conjecture that *Maximum Matching can be expressed in ESO Π_1 Horn Logic **if and only if** we can do the same for Maximum Independent Set and Maximum Clique.*

However, in Section 4, we will show that *Maximum Matching* cannot be expressed as an ESO Π_1 Horn sentence. (The inexpressibility of the other two problems in ESO Π_1 Horn can be proved similarly.)

In addition, we will show that another polynomially solvable decision problem, finding a solution for Vertex Cover within an approximation bound of two, cannot be expressed in ESO Π_1 Horn logic (this proof assumes that $\mathbf{P} \neq \mathbf{NP}$).

3 Expression for the Cardinality Constraint $|F| \geq K$

In this section, we derive an ESO expression for the cardinality constraint $|F| \geq K$. The construction given here was suggested by Neil Immerman [14].

L is a second order binary relation, from numbers to elements of the original domain A . That is, $L : N \rightarrow A$. Assume that $N = \{0, 1, 2, \dots, n\}$, where $n = |A|$, the number of vertices in the input graph.

$$\begin{aligned} \eta_1 &\equiv \forall y \forall v L(y, v) \rightarrow [F(v) \wedge num(y)] \equiv \forall y \forall v \neg L(y, v) \vee [F(v) \wedge num(y)] \\ &\equiv \forall y \forall v [\neg L(y, v) \vee F(v)] \wedge [\neg L(y, v) \vee num(y)]. \\ &\quad [L \text{ is a binary relation. For } L, \text{ the first argument } x \in N, \text{ and the second} \\ &\quad \text{argument } y \in F.] \\ \eta_2 &\equiv \forall v F(v) \rightarrow vertex(v) \\ &\equiv \forall y \neg F(v) \vee vertex(v). \quad [\text{as in (5), (10) and (14)}] \end{aligned}$$

L is not a function; it is possible that $L(x, u)$ and $L(x, v)$ are true when $u \neq v$. But the inverse of L is a function; given $u \in A$, it reverse maps to a unique x , in $L(x, u)$. That is,

$$\eta_3 \equiv \forall x \forall y \forall u [L(x, u) \wedge L(y, u)] \rightarrow (x = y). \quad (15)$$

(Every $u \in A$ has a unique index $x \in N$.)

Let the first order relations $succN$ and $succA$ define succession within N and A respectively. The Horn condition will **not** apply to these. Add the following 2 clauses to the expression of the cardinality constraint:

$$\eta_4 \equiv L(0, base) \wedge SuccA(base, min). \quad (16)$$

From here on, the forward recursion will take over and build all the correct L 's. The incorrect L 's will be established false by backward recursion.

L is a second order relation; it scans the entire domain A from $base$ to max ; as it scans A , it keeps a running count of the number of elements in A that are also members of F . (Note that $base$ has been assigned an index of zero in (16).)

Forward Recursion:

$$\begin{aligned} \eta_5 &\equiv \forall x \forall y \forall u \forall v [SuccN(x, y) \wedge SuccA(u, v) \wedge (u < max) \wedge L(x, u) \wedge F(v)] \rightarrow L(y, v) \\ &\equiv \forall x \forall y \forall u \forall v \neg SuccN(x, y) \vee \neg SuccA(u, v) \vee (u \geq max) \vee \neg L(x, u) \vee \neg F(v) \vee L(y, v). \end{aligned} \quad (17)$$

If $v \in F$, then u and v have different indices (x and y respectively). Otherwise, they share the same index (x).

$$\begin{aligned} \eta_6 &\equiv \forall x \forall y \forall u \forall v [SuccN(x, y) \wedge SuccA(u, v) \wedge (u < max) \wedge L(x, u) \wedge \neg F(v)] \rightarrow L(x, v) \\ &\equiv \forall x \forall y \forall u \forall v \neg SuccN(x, y) \vee \neg SuccA(u, v) \vee (u \geq max) \vee \neg L(x, u) \vee F(v) \vee L(x, v). \end{aligned} \quad (18)$$

Backward Recursion:

$$\begin{aligned}\eta_7 &\equiv \forall x \forall y \forall u \forall v [SuccN(x, y) \wedge SuccA(u, v) \wedge (v > min) \wedge L(y, v) \wedge F(v)] \rightarrow L(x, u) \\ &\equiv \forall x \forall y \forall u \forall v \neg SuccN(x, y) \vee \neg SuccA(u, v) \vee (v \leq min) \vee \neg L(y, v) \vee \neg F(v) \vee L(x, u).\end{aligned}\tag{19}$$

If $v \in F$, then u and v have different indices (x and y respectively). Otherwise, they share the same index (y).

$$\begin{aligned}\eta_8 &\equiv \forall x \forall y \forall u \forall v [SuccN(x, y) \wedge SuccA(u, v) \wedge (v > min) \wedge L(y, v) \wedge \neg F(v)] \rightarrow L(y, u) \\ &\equiv \forall x \forall y \forall u \forall v \neg SuccN(x, y) \vee \neg SuccA(u, v) \vee (v \leq min) \vee \neg L(y, v) \vee F(v) \vee L(y, u).\end{aligned}\tag{20}$$

The index of max is at least k :

$$\eta_9 \equiv \forall x L(x, max) \rightarrow (x \geq k) \equiv \forall x \neg L(x, max) \vee (x = k) \vee (x > k).\tag{21}$$

The first order variables in each of the nine η expressions can be renamed so that the variables in each expression is distinct. Thus the complete expression for the OFC is

$$\exists L \bigwedge_{i=1}^9 \eta_i.\tag{22}$$

As mentioned in Remark 19, the OFC expression above can be used for the three problems in Sec. 2. Again, as mentioned before, this is an ESO expression where the first order part is universal; however, it is *not* Horn, since expressions (18) and (20) are not.

4 Some PTIME decidable properties cannot be expressed as ESO Π_1 Horn sentences at the structure level

Under ordered structures, let \mathcal{P}_1 (\mathcal{P}_2) be the class of PTIME decidable properties that can be expressed in ESO Π_1 Horn logic at the machine (structure) level respectively.

In [7], it was shown that $\mathcal{P}_2 \subseteq \mathcal{P}_1 = \mathbf{P}$. Actually in [7], the author assumes the “successor relation” assumption, not ordered structures. However, if a property holds good under the former assumption, it also holds good under the latter (see Remark 15).

In this section, we show that there are properties in \mathbf{P} that cannot be expressed in ESO Π_1 Horn logic at the structure level. That is, we will prove that

Theorem 20. $\mathcal{P}_2 \subset \mathcal{P}_1$.

Theorem 20 can also be proven using the “preservation under substructure” property mentioned in Remark 15. However, we will give two other proofs as follows:

- (i) One using reduced products, by showing that the decision version of Maximum Matching (a PTIME decidable property) cannot be expressed in ESO Π_1 Horn at the structure level;
- (ii) Another using approximability theory as applied to the Vertex Cover problem. This proof assumes that $\mathbf{P} \neq \mathbf{NP}$. The question of whether there exists a feasible solution to a Vertex Cover instance, whose value is within a factor of two of the optimal solution value, is PTIME decidable.

4.1 Proof based on reduced products

We show that Maximum Matching (decision version) lies in $\mathcal{P}_1 \setminus \mathcal{P}_2$, which will prove Theorem 20.

The proof here is adapted from the one given by Dawar [3], which showed that the cardinality constraint $|F| \geq K$ cannot be expressed by a Horn formula, when F is an existentially quantified second-order predicate. F is unknown; it is not a part of the input. Recall that K is the lower bound on the objective function value.

The proof relies on the fact that universal Horn formulae are preserved under reduced products. That is, if ψ is a universal Horn formula and structures \mathbf{A} and \mathbf{B} satisfy ψ , then so does their reduced product $\mathbf{A.B}$.

Lemma 21. *Maximum Matching (decision version) cannot be expressed in ESO Π_1 Horn at the structure level, and hence this problem is not a member of \mathcal{P}_1 .*

Proof. Suppose we have a signature divided into two parts:

A set of relation symbols $\Sigma = \{S_1, \dots, S_m, F\}$, and a separate set ρ of relation and constant symbols.

The relations in ρ are first order (input), hence the Horn restriction does not apply to these. On the other hand, the relations in Σ are second order (quantified), and hence the Horn condition applies to these relations. In particular, it applies to F .

Let \mathbf{A} and \mathbf{B} be two input structures on the same universe $\{1, \dots, 2\theta\}$ ($\theta \in \mathbb{N}$) with a linear ordering; furthermore, for all symbols in the signature ρ , let \mathbf{A} and \mathbf{B} agree on the interpretation of the symbols; however, they may differ on relations in Σ .

Let us now define a product structure (a *reduced product*) $\mathbf{A.B}$ to be the structure on the domain $\{1, \dots, 2\theta\}$ with a linear ordering, in which every symbol in the signatures Σ and ρ is interpreted as the *intersection* of the two interpretations in \mathbf{A} and \mathbf{B} . Then it is easy to show that any Horn formula ϕ that is satisfied in both \mathbf{A} and \mathbf{B} is also satisfied in $\mathbf{A.B}$. (This is exactly by the argument that shows that Horn formulae are closed under direct products; for example, see Chapter 9 (Page 417) of the book by Wilfrid Hodges [10], or Page 493 of his book chapter [11].)

Then suppose we let \mathbf{A} and \mathbf{B} be structures with, say, 2θ elements $\{1, \dots, 2\theta\}$; In \mathbf{A} , we let F consist of the first θ elements $\{1, \dots, \theta\}$; and in \mathbf{B} , we let F consists of the last θ elements $\{\theta + 1, \dots, 2\theta\}$; Also, we let $K = \theta$ in both structures. So, the property “ $|F| \geq K$ ” is true in both \mathbf{A} and \mathbf{B} . That is, there exists a matching of size at least θ in \mathbf{A} and \mathbf{B} .

However, “ $|F| \geq K$ ” is false in $\mathbf{A.B}$, since in this structure, F is empty, even though the lower bound K is still equal to θ .

It follows that the constraint $|F| \geq K$ cannot be expressed as a Horn formula, and hence we conclude the decision version of Maximum Matching, a PTIME solvable problem, cannot be expressed in ESO Π_1 Horn at the structure level. \square

This proves Theorem 20.

4.2 Proof based on approximation theory

Here we provide another proof of Theorem 20, under the assumption that $\mathbf{P} \neq \mathbf{NP}$. We use approximation theory to show that some decision problems derived from \mathbf{NP} -hard optimization problems cannot be expressed in ESO Π_1 Horn logic. We need a few definitions first.

Given an NP-optimization problem Q (see Def. 3), we define the following:

Definition 22. Approximation ratio. *Given an instance I of Q and a feasible solution $S \in F_Q(I)$, the approximation ratio is defined as $\frac{f_Q(I, S)}{f_Q^{(opt)}(I)}$ for minimization problems and $\frac{f_Q^{(opt)}(I)}{f_Q(I, S)}$ for maximization problems. We assume that $f_Q^{(opt)}(I) > 0$, where $f_Q^{(opt)}(I)$ is the optimal solution value to instance I .*

Definition 23. Approximation problems A_Q^T and A_Q^B .

Problem A_Q^T . Given an instance I of problem Q and a parameter K_Q that depends on Q , is there a feasible solution $S \in F_Q(I)$, such that the approximation ratio is at most K_Q ?

Problem A_Q^B . Given Q , I and a parameter L_Q that depends on Q , is there a feasible solution $S \in F_Q(I)$, such that the approximation ratio is at most L_Q ?

($L_Q < K_Q$.)

The APX class of \mathbf{NP} -hard optimization problems is defined as follows:

Definition 24. The APX class. *Consider a problem Q from the class \mathbf{NP}_{opt} . Then Q is a member of the class APX iff the decision problem A_Q^T can be solved in PTIME.*

Of course, if A_Q^T can be solved in PTIME and $K_Q = 1$, then $Q \in \mathbf{P}_{opt}$.

Example: Steiner Tree is a problem in APX, with parameter $K_Q = 2$. The optimization version of this problem can be described as follows:

Given: A graph $G = (V, E)$, a subset S of V and edge weights $w : E \rightarrow \mathbb{N}$.

To Do: Find a subgraph T of G such that

- (i) *The BFC: T includes all the vertices in S , T is a tree, and*
- (ii) *The objective function: The sum of the weights of the edges in T is to be minimized.*

Hence given an instance I of the Steiner tree problem, there is a PTIME algorithm which can decide whether there is a feasible solution to I with an approximation ratio of at most two.

However, K_Q is an *upper bound* on the approximation ratio; we know that there is a PTIME algorithm that guarantees a feasible solution within this upper bound. However, many problems (such as Vertex Cover, below) in the class APX also have a proven *lower bound* L_Q . For such problems, the approximation problem A_Q^B is known to be \mathbf{NP} -complete.

Definition 25. Vertex Cover (optimization version).

Given: A graph $G = (V, E)$.

To do: Mark vertices in V in such a way that

- (i) *BFC: For any $x, y \in V$ where $x < y$, if $(x, y) \in E$, then either x or y is marked, and*
- (ii) *The objective function: The number of marked vertices is to be minimized.*

For Vertex Cover, the upper bound $K_{vc} = 2$ and the lower bound $L_{vc} = 7/6$ (see [9])⁸. In fact, for L_{vc} , one could use any number γ such that $1 < \gamma \leq 7/6$. For ease of representation in an ESO formula, let us use $\gamma = 1.1$.

Let A_{vc}^T and A_{vc}^B be the approximation problems \mathbf{A}_Q^T and \mathbf{A}_Q^B respectively (from Def. 23), applied to Vertex Cover.

A_{vc}^T and A_{vc}^B are members of the class \mathbf{NP} ; in particular, A_{vc}^T is a member of the class \mathbf{P} , and A_{vc}^B is \mathbf{NP} -complete.

Then it follows that A_{vc}^T can be expressed as an ESO Π_1 Horn sentence ϕ_1 . This sentence will include a sub-formula stating that the approximation ratio $\alpha(I, S)$ is at most two ($\alpha(I, S) \leq 2$).

However, if we replace the formula (for the condition) $\alpha(I, S) \leq 2$ with the formula for $\alpha(I, S) \leq \gamma = 1.1$, we obtain a new ESO Π_1 Horn sentence ϕ_2 which expresses A_{vc}^B .

Though A_{vc}^B is known to be \mathbf{NP} -complete, we have just obtained an ESO Π_1 Horn sentence (ϕ_2) for this problem, implying that A_{vc}^B can be decided in PTIME. (See Theorem 7.) This contradicts our assumption that $\mathbf{P} \neq \mathbf{NP}$.

(If $\mathbf{P} = \mathbf{NP}$, then $K_Q = L_Q = 1$.)

Hence we have shown the following:

Lemma 26. *Assuming that $\mathbf{P} \neq \mathbf{NP}$, A_{vc}^T , a problem solvable in PTIME, cannot be expressed as an ESO Π_1 Horn sentence at the structure level; that is, $A_{vc}^T \in \mathcal{P}_1 \setminus \mathcal{P}_2$.*

This proves Theorem 20, under the assumption that $\mathbf{P} \neq \mathbf{NP}$.

Here is another example:

Linear Programming (LP). In the case of the decision version of LP, every constraint, whether OFC or BFC, is of the form $\sum_j a_{ij}x_j \leq b_i$ or $\sum_j a_{ij}x_j \geq b_i$ (where the a_{ij} 's and the b_i 's are constants), that is, checking the cardinality of the summation on the left side. From Remark 15, we know that such constraints cannot be expressed in ESO Π_1 Horn form at the structure level. Hence LP is not a member of the class \mathcal{P}_2 . This observation can be generalized to other Mathematical Programming problems such as Convex Programming where the constraints are of a more general form such as $f(\mathbf{x}) \leq K$ (where K is a constant).

4.3 $|F| \leq K$ cannot be expressed as a universal Horn formula

Previously we saw that the lower bound $|F| \geq K$ cannot be represented in ESO Π_1 Horn. We can show a similar inexpressibility result for the upper bound ($|F| \leq K$) using direct products.

The proof relies on the fact that universal Horn formulae are preserved under direct products. That is, if ψ is a universal Horn formula and structures \mathbf{A} and \mathbf{B} satisfy ψ , then so does their direct product $\mathbf{A} \times \mathbf{B}$.

Consider identical input structures \mathbf{A} and \mathbf{B} , with domain as $\{1, 2, \dots, 2\theta\}$. Let every relation in \mathbf{A} and \mathbf{B} be interpreted in the same way, including F (the objective function relation).

⁸Johan Håstad has been awarded the 2011 Gödel Prize for this publication.

Let $F^{(A)} = F^{(B)} = \{1, \dots, \theta\}$. Hence $|F^{(A)}| = |F^{(B)}| = \theta$. Let $K = \theta > 1$.

In the direct product $\mathbf{A} \times \mathbf{B}$, the domain $|\mathbf{A} \times \mathbf{B}| = \{(1, 1), \dots, (2\theta, 2\theta)\}$.

Then $F^{(A \times B)} = \{(1, 1), \dots, (\theta, \theta)\}$. That is, $|F^{(A \times B)}| = \theta^2 > K$.

Hence the upper bound on the objective function, that $|F| \leq K$, is not preserved in the product structure. We can conclude that

Theorem 27. *The upper bound form of the OFC, $|F| \leq K$, cannot be captured in ESO Π_1 Horn logic.*

This would be applicable to decision problems in \mathbf{P} that are derived from minimization problems.

5 Implications for the class \mathbf{NP} and Fagin's theorem

Under ordered structures, let \mathcal{P}_3 (\mathcal{P}_4) be the class of \mathbf{NP} -decidable properties that can be expressed in ESO logic where the first order part is a universal formula at the machine (structure) level respectively.

It is known that $\mathcal{P}_4 \subseteq \mathcal{P}_3 = \mathbf{NP}$. See Chapter 7 in [13] or Chapter 3 in [8].

However, no problem in \mathbf{NP} derived from a maximization problem can be expressed with a universal first order part, since the OFC cannot be so expressed.

For the problem as such (considering the BFC and the OFC), it will violate the “preservation under substructure” property of universal formulae. For a given structure A and lower bound K , if it satisfies the BFC and the OFC, we can consider a substructure that is small enough which will violate the OFC. It follows that

Theorem 28. $\mathcal{P}_4 \subset \mathcal{P}_3$.

It is still open whether Fagin's theorem (that ESO logic characterizes the class \mathbf{NP}) holds at the structure level when the input structures have no ordering.

6 Structure level and least fixed point

The Immerman-Vardi theorem (see [8], page 70), which states that least fixed point logic (LFP) characterizes \mathbf{P} for the class of ordered finite structures, is true at the machine level. We now investigate whether it holds good at the structure level.

Using results from Atserias et al [2], we can show that least fixed point logic with counting (LFP+C) is also insufficient to capture PTIME at the structure level. Consider the following problem:

Definition 29. FEASIBLE. *Given a matrix $A \in \mathbb{Z}_2^{m \times n}$ and a vector $b \in \mathbb{Z}_2^m$, does the system of equations $AX = b$ have a solution $X \in \mathbb{Z}_2^n$?*

Under ordered structures, let \mathcal{P}_5 (\mathcal{P}_6) be the class of PTIME decidable properties that can be expressed in LFP+C at the machine (structure) level respectively.

Theorem 30. (*Immerman-Vardi theorem*) $\mathcal{P}_5 = \mathbf{P}$.

$\mathbf{C}_{\infty\omega}$ is the logic obtained from atomic formulae with negation, infinitary conjunctions and disjunctions and counting quantifiers $\exists^{\geq i}$, where i is a non-negative integer. $\mathbf{C}_{\infty\omega}^k$ is a fragment of $\mathbf{C}_{\infty\omega}$ with only k distinct variables in the formula. $\mathbf{C}_{\infty\omega}^\omega$ is defined as:

$$\mathbf{C}_{\infty\omega}^\omega = \bigcup_{k \in \omega} \mathbf{C}_{\infty\omega}^k. \quad (23)$$

FEASIBLE can be decided in PTIME [2]. It was also shown in [2] that FEASIBLE is not expressible in $\mathbf{C}_{\infty\omega}^\omega$. LFP+C is a fragment of $\mathbf{C}_{\infty\omega}^\omega$; any property that can be expressed in LFP+C can also be expressed in $\mathbf{C}_{\infty\omega}^\omega$.

Since FEASIBLE, a problem in \mathbf{P} , cannot be expressed in $\mathbf{C}_{\infty\omega}^\omega$, it cannot be expressed in LFP+C either. From this, we conclude that LFP+C is insufficient to capture PTIME at the structure level. That is,

Theorem 31. $\mathcal{P}_6 \subset \mathcal{P}_5$.

7 Conclusions

In the last 40 years, we have seen important contributions that provided machine level expressions [5, 12]. However, as mentioned in Sec. 1.2, Descriptive Complexity is a much more useful tool at the structure level than at the machine level.

For a large subset of the class \mathbf{P} (those problems derived from optimization), the usefulness of Descriptive Complexity is lost when it comes to recognizing their PTIME decidability using expressibility in ESO Π_1 Horn logic.

In the case of ESO Π_1 Horn logic, we were able to demonstrate a separation between machine level and the structure level — the machine level can capture \mathbf{P} , whereas the structure level cannot. Hence we need to investigate other logics that can close this gap. The inexpressibility proofs in Section 4 also hold good when the input structures are ordered.

The Immerman-Vardi theorem is true at the machine level, but not at the (more important) structure level. Also at the structure level, it is still open whether Fagin's theorem holds when the input structures are unordered.

Another interesting thing we have observed is, the BFC expressions for some \mathbf{NP} -complete problems are exactly identical to those for some problems in \mathbf{P} (Remark 18). Considering this fact and Remark 19 about the OFC, we conclude that ESO universal sentences are unable to distinguish between \mathbf{P} and \mathbf{NP} -complete problems at the structure level.

8 Acknowledgments

Anuj Dawar (Cambridge), Wilfrid Hodges (Queen Marys, London), Neil Immerman (Massachusetts), Leonid Libkin (Edinburgh) and Sanming Zhou (Melbourne).

References

- [1] S. Abiteboul and M.Y. Vardi and V. Vianu. Fixpoint logics, relational machines and computational complexity. *Journal of the ACM*, 44(1):30–56, January 1997.
- [2] A. Atserias and A. Bulatov and A. Dawar. Affine systems of equations and counting infinitary logic. *Theoretical Computer Science*, 410(18):1666–1683, April 2009.
- [3] A. Dawar. Personal communication, December 2009.
- [4] H.D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer, 1999.
- [5] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In R. Karp, editor, *Complexity of Computations*, pages 43–73. SIAM-AMS Proceedings (no.7), 1974.
- [6] J. Gate and I.A. Stewart. Frameworks for logically classifying polynomial-time optimization problems. In F. Ablayev and E.W. Mayr, editors, *Proceedings of the Fifth International Computer Science Symposium in Russia (CSR 2010) — Lecture Notes in Computer Science*, volume 6072, pages 120–131. Springer, June 2010.
- [7] E. Grädel. The expressive power of second order Horn logic. In *STACS 1991: Proceedings of the 8th annual symposium on Theoretical aspects of computer science — Lecture Notes in Computer Science 280*, pages 466–477. Springer-Verlag, 1991.
- [8] E. Grädel, P.G. Kolaitis, L. Libkin, M. Marx, J. Spencer, M.Y. Vardi, Y. Venema, and S. Weinstein. *Finite Model Theory and Its Applications*. Texts in Theoretical Computer Science. Springer, 2007.
- [9] J. Håstad. Some Optimal Inapproximability Results. *Journal of the ACM*, 48(4):798–859, July 2001.
- [10] W. Hodges. *Model Theory*. Cambridge University Press, 1993.
- [11] W. Hodges. Logical Features of Horn Clauses. In Dov M. Gabbay, C.J. Hogger and J.A. Robinson (eds.), *Handbook of Logic in Artificial Intelligence and Logic Programming (Vol. 1)*, pages 449–503. Oxford University Press (NY), 1993.
- [12] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68(1-3):86–104, 1986.
- [13] N. Immerman. *Descriptive Complexity*. Springer-Verlag, 1999.
- [14] N. Immerman. Personal communication, September 2009.
- [15] N.D. Jones and W.T. Laaser. Complete problems for deterministic polynomial time. *Theoretical Computer Science*, 3(1):105–117, October 1976.
- [16] P. Manyem. Syntactic Characterizations of Polynomial Time Optimization Classes. *Chicago Journal of Theoretical Computer Science*, 3, 2008.
- [17] A. Panconesi and D. Ranjan. Quantifiers and approximation. *Theoretical Computer Science*, 107:145–163, 1993.

- [18] H. Whitney. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, 54 (1): 150-168, 1932.